UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/846,067	04/30/2001	Rahul Sharma	SUNMP007	4516

25920 7590 06/16/2004

MARTINE & PENILLA, LLP
710 LAKEWAY DRIVE
SUITE 170
SUNNYVALE, CA 94085

EXAMINER	
PHAM, CHRYSTINE	
ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 06/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PRG

Office Action Summary	**Application No.** 09/846,067	**Applicant(s)** SHARMA ET AL.	
	Examiner Chrystine Pham	**Art Unit** 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)□ Responsive to communication(s) filed on _____.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 April 2001* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Drawings

1. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because:

   - They do not include the following reference character(s) mentioned in the description: 304b (Fig.9 operation 919 – pg.29 line 11)
   - They include the following reference character(s) not mentioned in the description: 608/dependentObject (Fig.6), 715 (Fig.7), and 917 (Fig.9).

2. The drawings are objected to under 37 CFR 1.83(a) because they fail to show operations as described in the specification. For example, operation 713 (see Fig.7) labels an "upgrade Module" operation and an arrow, which starts from 750b/ControlModule and ends at 202/Executive. However, the specification states that "... new state object 750b upgrades all child J2EE Servers, in operation 713". Similar issue is present in operation 714 (see Fig.7 and pg.26 line 11-12). Any structural detail that is essential for a proper understanding of the disclosed invention should be shown in the drawing. MPEP § 608.02(d).

3. Corrected drawing sheets, or amendment to the specification to add the reference character(s) in the description, are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to

obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The disclosure is objected to because of the following informalities:

- Cross-reference to related applications does not contain updated status of the applications (application numbers);
- Inconsistent use of "RSM 204" (pg.14 line 22) since RSM is previously declared as Replicated State Manager and 204 previously refers to J2EE Server.

Appropriate correction is required.

5. The use of the trademark JAVA has been noted in this application (e.g., claim 1 line 1, and claim 6 line 1). It should be capitalized wherever it appears and be accompanied by the generic terminology. Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner, which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> *Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.*

Double Patenting

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed.

Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

8. Claims 1-18 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims **1-13** of copending Application No. 09833845 (hereinafter *copending application*). Although the conflicting claims are not identical, they are not patentably distinct from each other because both sets of claims are directed to a method of performing an online upgrade of a Java application.

This is a <u>provisional</u> obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

As per claim 1, the *copending application* claims a method for performing an online upgrade of a Java application (claim **1**, line 1), the method comprising:

- o Executing a Java module on a server, the Java module including at least one original entity bean (see *original service module,* claim **1**, line 3) and at least one original state object (see *original control module,* claim **1**, line 3-4) in communication with the original entity bean, the original state object storing a state of the original entity bean (see *application-specific policies* claim **1**, line 4).

- Generating an upgraded state object (see *upgraded control module* claim **1**, line 6).
- Transferring the state stored in the original state object to the upgraded state object (claim **7**, line 1-2).
- Providing state management for the original entity bean using the upgraded state object (claim **1**, line 7).

As per claims 2-3, see *creating upgraded service module using the upgraded control module* (claim **1** line 7-8).

As per claim 4, the *copending application* claims a method as applied to claim 3, wherein both the original entity bean and the original state object are disabled (claim **4**, line 1-2).

As per claim 5, the *copending application* claims a method as applied to claim 1, wherein the upgraded state object is generated by upgrading the physical schema, which <u>contains state object classes</u> (emphasis added), using data stored in a repository (claim **2**, line 1-3).

As per claim 6, the *copending application* claims a method as applied to claim 5, wherein functionality of the Java module is not disrupted when the upgraded state object is generated (claim **5** line 1-2).

As per claim 7, it recites limitation, which has been addressed in claim 6 above, therefore, is rejected for the same reason as cited in claim 6.

As per claim 8, the *copending application* claims a Java platform capable of performing an online upgrade on a Java application (claim **8** line 1), the Java platform comprising:

- A Java module including at least one original entity bean and at least one original state object (see *original control module*, claim **8** line 3) in communication with the original entity bean (see *original service module*, claim **8** line 3), wherein the original state object storing a state of the original entity bean (see *application-specific policies* claim **8**, line 4), and wherein the state object provides state management for the original entity bean (claim **8** line 4-5).

- A repository having upgraded class files for the original entity bean and upgraded class files for the original state object (claim **8** line 6-7).
- Wherein the original state object is upgraded by generating an upgraded state object using upgraded class files from the repository (claim **8** line 8-10), and transferring the state stored in the original state object to the upgraded state object (claim **13** line 1-3).

As per claim 9, the *copending application* claims a method as applied to claim 8, wherein an upgraded entity bean is created when upgrading the Java platform (claim **9** line 1-3).

As per claim 10, the *copending application* claims a method as applied to claim 9, wherein the state of the upgraded entity bean is managed using the upgraded state object (claim **8** line 10-11).

As per claim 11, see claim **10**.

As per claim 12, the *copending application* claims a method as applied to claim 8, wherein the upgraded state object is generated by upgrading the physical schema, which contains state object classes (emphasis added), using data stored in a repository (claim **8** line 8-10).

As per claim 13, the *copending application* claims a method as applied to claim 12, wherein functionality of the Java module is not disrupted when the upgraded state object is generated (claim **11** line 1-2).

As per claim 14, it recites limitation, which has been addressed in claim 13 above, therefore, is rejected for the same reason as cited in claim 13.

As per claim 15, the *copending application* claims a method for upgrading a Java application having a managed application state (claim **8** line 1-4) comprising the operations of:

- Executing a Java module on a server, the Java module including at least one original entity bean (see *original service module*, claim **8** line 3) and at least one original state object (see *original control module*, claim **8** line 3) in communication with the original entity bean, the original state object storing the state of the original entity bean (see *application-specific policies* claim **8**, line 4).
- Generating an upgraded state object using data stored in a system repository (claim **8** line 8-10).
- Transferring the state stored in the original state object to the upgraded state object (claim **13** line 1-3).
- Providing state management for the original entity bean using the upgraded state object (claim **8** line 10-11).
- Generating an upgraded entity bean using data stored in a system repository (claim **9** line 1-3).
- Providing state management for the upgraded entity bean using the upgraded state object (claim **8** line 10-11).
- Disabling both the original entity bean and the original state object (claim **10** line 1-2).

As per claims 16-18, they recite limitations, which have been addressed above in claims 12-14, respectively, therefore, are rejected for the same reasons as cited in claims 12-14 from above.

9. Claims 19-20 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims **1-13** of the *copending application* in view of Anderson Jesper (XP-002249737) of record (hereinafter *Anderson*).

This is a provisional obviousness-type double patenting rejection.

As per claim 19, the *copending application* claims a method as applied to claim 18. The *copending application* fails to teach the original state object and the upgraded state object being classified into a particular state management unit. However, *Anderson* discloses a method for online upgrade of a Java application, wherein both the original state object and the upgraded state object are classified into a particular state management unit (e.g., see section Strategy, col.4 line 51 – col.5 line 8). It would have been obvious to one of ordinary skill in the pertinent art at the time of the applicant's invention to modify the teaching of the *copending application* to include a classification of the original and upgraded state objects into a state management unit as disclosed by *Anderson*, for said classification will enable migration of components through updates, preserve data consistency and transparency of the online upgrade.

As per claim 20, the *copending application*, as modified by *Anderson*, claims a method as applied to claim 19, wherein the particular state management unit is used to facilitate upgrading of the original state object (e.g., see section Strategy, col.5 line 16-18).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at*

> *the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

11. Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ma et al. (U.S. Patent 5,920,725) made of record (hereinafter *Ma et al.*).

As per claim 1, *Ma et al.* teach a method for upgrading managed state for a Java application (e.g., see Abstract), the method comprising:

- Executing Java module having at least one original entity bean and at least one original state object in communication with the original entity bean (e.g., FIG.5 *server app 86 & client app 74* & associated text), the original state object storing a state of the original entity bean (e.g., see FIG.5 *rules 81*, col.8 line 37-39).
- Generating an upgraded state object (e.g., col.8 line 55, FIG.3 classes 68', 68 & associated text).
- Transferring the state stored in the original state object to the upgraded state object (e.g., col.9 line 20-27, col.11 line 25-40).
- Providing state management for the original entity bean using the upgraded state object (e.g., FIG.8 *152* & *144* & associated text).

As per claims 2-3, they recite limitations that have been previously addressed in claim 1 above. Therefore, they are rejected for the same reasons as cited in claim 1.

As per claim 4, *Ma et al.* disclose a method as applied to claim 3, wherein both the original entity bean and the original state object are disabled (e.g., col.4 line 59-63, col.5 line 17-21).

As per claim 5, *Ma et al.* disclose a method as applied to claim 1, wherein the upgraded state object is generated by upgrading the physical schema, which contains state object classes (emphasis added), using data stored in a repository (e.g., see FIG.3 *repository 62* and associated text, col.4 line 42-48).

As per claim 6, the *Ma et al.* claims a method as applied to claim 5, wherein functionality of the Java module is not disrupted when the upgraded state object is generated (e.g., see Abstract, col.7 line 41-43, col.10 line 55-56, and also col.4 line 59-63).

As per claim 7, it recites limitation, which has been addressed in claim 6 above, therefore, is rejected for the same reason as cited in claim 6.

As per claim 8, the *Ma et al.* claims a Java platform capable of performing an online upgrade on a Java application, the Java platform comprising:

- A Java module including at least one original entity bean and at least one original state object in communication with the original entity bean (e.g., FIG.5 *server app 86 & client app 74* & associated text), wherein the original state object storing a state of the original entity bean, and wherein the state object provides state management for the original entity bean (e.g., see FIG.5 *rules 81*, col.8 line 37-39).
- A repository having upgraded class files for the original entity bean and upgraded class files for the original state object (e.g., see FIG.3 *repository 62* and associated text, col.4 line 42-48).
- Wherein the original state object is upgraded by generating an upgraded state object using upgraded class files from the repository (e.g., see FIG.3 *repository 62* and associated text, col.4 line 42-48), and transferring the state stored in the original state object to the upgraded state object (e.g., col.9 line 24-33, col.11 line 25-40).

As per claim 9, the *Ma et al.* claims a method as applied to claim 8, wherein an upgraded entity bean is created when upgrading the Java platform (e.g., col.7 line 19-39,46-48, col.6 line 19-23, col.9 line 20-22).

As per claim 10, the *Ma et al.* claims a method as applied to claim 9, wherein the state of the upgraded entity bean is managed using the upgraded state object (e.g., see FIG.8 *146 or 147* & *152* & associated text).

As per claims 11-14, they recite limitations that have been previously addressed in the above claims 4-7 respectively, therefore, are rejected for the same reason as cited in claims 4-7.

As per claim 15, the *Ma et al.* claims a method for upgrading a Java application having a managed application state comprising the operations of:

- Executing a Java module on a server, the Java module including at least one original entity bean and at least one original state object in communication with the original entity bean (e.g., FIG.5 *server app 86* & *client app 74* & associated text), the original state object storing the state of the original entity bean (e.g., FIG.5 *rules 81*, col.8 line 37-39).
- Generating an upgraded state object using data stored in a system repository e.g., see FIG.3 *repository 62* and associated text, col.4 line 42-48).
- Transferring the state stored in the original state object to the upgraded state object (e.g., col.9 line 24-33, col.11 line 25-40).
- Providing state management for the original entity bean using the upgraded state object (e.g., see FIG.8 *144* & *152* and associated text).

- Generating an upgraded entity bean using data stored in a system repository (e.g., col.4 line 45-48, col.6 line 12-15).
- Providing state management for the upgraded entity bean using the upgraded state object (e.g., see FIG.8 *146* or *147* & *152* and associated text).
- Disabling both the original entity bean and the original state object (e.g., see Abstract and also col.11 line 49-55).

As per claims 16-18, they recite limitations, which have been addressed above in claims 12-14, respectively, therefore, are rejected for the same reasons as cited in claims 12-14 from above.

12. Claims 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Ma et al.* as applied to claims 15-18 above, and further in view of *Anderson*.

As per claim 19, the Ma et al. disclose a method as applied to claim 18. Ma et al. fail to teach the original state object and the upgraded state object being classified into a particular state management unit. However, *Anderson* discloses a method for online upgrade of a Java application, wherein both the original state object and the upgraded state object are classified into a particular state management unit (e.g., see section Strategy, col.4 line 51 – col.5 line 8). It would have been obvious to one of ordinary skill in the pertinent art at the time of the applicant's invention to modify the teaching of Ma et al. to include a classification of the original and upgraded state objects into a state management unit as disclosed by *Anderson*, for said classification will enable migration of components through updates, preserve data consistency and transparency of the online upgrade.

As per claim 20, the teaching of Ma et al., as modified by *Anderson*, discloses a method as applied to claim 19, wherein the particular state management unit is used to facilitate upgrading of the original state object (e.g., see section Strategy, col.5 line 16-18).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Method for instantiating a class having different versions, Reich et al. (U.S. Patent 6,175,855).
- Live upgrade process for object-oriented programs, Moser et al. (U.S. Patent 6,360,363).
- Method and system for automatic detection and distribution of code version updates, Lam et al. (U.S. Patent 6,272,677).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 703.605.1219. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on 703.305.4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chrystine Pham
Examiner

TUAN DAM
SUPERVISORY PATENT EXAMINER